BUAA Intensive Mini Course

Intensive Training in High-performance Computing

Program

Beijing Advanced Innovation Center for Big Data and Brain Computing
School of Mathematical Sciences
Beihang University, Beijing, China
December 16-20, 2019

General Information

Organizers

Dongming Wang Chenqi Mou

Instructor

Ilias S. Kotsireas

Wilfrid Laurier University, Waterloo, Ontario, CANADA

Time and Place

December 16-20, 2019

Morning: 9:00-11:30, Room F222, New Main Building

Afternoon: 14:00-16:30, Room 309, Building No. 3

Meals

Lunches (Dec 16-20): Training Center Restaurant

Contact

Dong Lu, tel: +86-1788 8817 132, email: donglu@amss.ac.cn

Juan Xu, tel: +86-1340 1187 261, email: xujuan0505@126.com

Program

The course is structured in 10 units. Each unit will consist of a 2-hour session during which there will be some discussion of theoretical concepts which will be then illustrated by fully worked out examples on the computer.

Course Schedule		
Dec 16	Morning	Linux: basic commands, environment and tools, (sed, awk)
	Afternoon	Bash shell: basic commands, programming and scripting
Dec 17	Morning	Introduction to Maple: basic commands packages, functions, procedures, programming
	Afternoon	Introduction to meta-programming: generating bug-free code automatically
Dec 18	Morning	Parallel computing with Message Passing Interface (MPI): basic principles, code examples
	Afternoon	Combinatorial computing: basic tools and techniques, examples
Dec 19	Morning	Problem solving on a supercomputer I
	Afternoon	Problem solving on a supercomputer II
Dec 20	Morning	Problem solving on a supercomputer III
	Afternoon	Problem solving on a supercomputer IV

Introduction

The purpose of this intensive training course in high-performance computing is to supply the participants with the necessary skills to be able to solve very hard combinatorial problems on a supercomputer.

The approach that I advocate is to use meta-programming techniques to produce bug-free and provably correct code. The participants will learn a number of tools that can be used to implement my approach. Another advantage of using meta-programming techniques is that they allow one to produce fast and reliably several millions of lines of code that can then be compiled and executed on a supercomputer.

The approach will be illustrated with several problem solving examples on an actual supercomputer. Participants will be assigned projects to carry out on a supercomputer. Participants are also encouraged to come forward with their own projects of interest, if these projects require significant computational resources.

The targeted audience of this intensive mini course is PhD students, post-doctoral fellows, faculty members and staff members that have never been exposed to working with a supercomputer before and that are interested to tame the immense computational power of modern supercomputers to solve extremely hard combinatorial problems.

Prerequisites

The course is self-contained and does not require any prior knowledge of combinatorial computing, supercomputing, Linux, Maple and meta-programming. There is a notable exception, in that participants should have a working knowledge of the C programming language, in order to benefit the most from the course. I will not cover the C programming language due to lack of time. However, i will be available if participants wish to ask me questions regarding the C programming language. Although some more esoteric features of C will be explained whenever these are needed, the participants are expected to be familiar with the C programming language. Participants will be required to have access to a Linux server that they can login to, using for instance free software tools such as PuTTY http://www.putty.org/ or MTPuTTY http://ttyplus.com/ or any other ssh client of their choice.

Instructor

The course will be taught by Dr. Ilias S. Kotsireas, a Professor of Computer Science at Wilfrid Laurier University, in Waterloo, Ontario, Canada. The instructor has 15 years of experience in high-performance computing and has been conducting research using supercomputing facilities in Canada, China, Japan, Spain and the United States. His research interests are in Design Theory, Coding Theory, Cryptography, High-performance Computing and Computational Algebra. He has over 100 published journal and conference papers, edited books, technical reports, chapters in books and handbooks and special issues of journals. His research has been funded by NSERC for 15 years. He has organized a large number of international conferences in North America, Europe and Asia.

He currently serves as the Director of the CARGO Lab, http://www.cargo.wlu.ca/

He currently serves as the Chair of ACM SIGSAM, http://www.sigsam.org/

Training materials, contact and resources

The slides of all presentations will be made available to all participants in electronic form. Attendees are free to use these materials privately and also share them with direct collaborators. I would request that no materials are publicly disseminated without explicit written authorization from the course Instructor. The instructor can be contacted via email at **ikotsire@wlu.ca** Several additional resources for further study will be presented during the lectures. These include books, papers, websites, code repositories and so forth.

The Essential Tool for Mathematics: Maple

Maple^{TM} is math software that combines the world's most powerful math engine with an interface that makes it extremely easy to analyze, explore, visualize, and solve mathematical problems. With Maple, you aren't forced to choose between mathematical power and usability, making it the ideal tool for both education and research. Maple helps you to enrich your classroom & accelerate your research by:

- Demonstrate and explore concepts using intuitive Clickable Math techniques, taking advantage of hundreds of tools specially designed for teaching and learning mathematics.
- Create course notes, assignments, interactive apps, and research reports that include live computations, plots, and mathematics.
- Solve advanced problems and develop algorithms with sophisticated, math-aware commands and programming language.

Application Areas Include: Calculus, Algebra, Matrices and Vectors, Visualization, Statistics, Geometry, Differential Equations, Polynomial Systems, Advanced Mathematics Control Design, Physics, Optimization, Financial Modeling, Scientific Data Management, Signal Processing, Transforms, Curve Fitting, Units and Tolerances, Code Generation, Application Development, Parallel and Grid Computing, Web Deployment and much more!

Maple's Extremely Powerful Math Engine has the depth, breadth, and performance to meet all your mathematical challenges:

- Over 5000 functions covering virtually every area of mathematics, including calculus, algebra, differential equations, statistics, linear algebra, geometry, and much more

- Symbolic, numeric, and hybrid computation algorithms
- World-leading algorithms for solving problems
- Sophisticated 2D and 3D plotting and animations
- Efficient algorithms and tools for high performance computing and large-scale problem solving

Maple is incredibly easy to use. Whether you are doing a quick calculation, developing complex algorithms, illustrating a concept, or creating an interactive technical document, Maple makes it easy to get the job done.

- Clickable Math interaction for point-and-click problem solving
- Sophisticated programming language designed for mathematics
- Specialized tools specifically for teaching and learning key topics in calculus, algebra, and more
- Rich authoring environment for creating technical documents and applications